1. **Write a Java program to print the string data in initcap format?**

```
public class SampleTest {

    public static void main(String[] args) {
        String input = "hello world from java 8";
        String result = Arrays.stream(input.split(" "))
                              .map(word -> word.substring(0, 1).toUpperCase() + word.substring(1))
                              .collect(Collectors.joining(" "));
        System.out.println(result); // Output: Hello World From Java 8

    }
```

2. **How do you find frequency of each character in a string using Java 8 streams?**

```
public class SampleTest {

    public static void main(String[] args) {
        String inputString = "Java Concept Of The Day";

        Map<Character, Long> charCountMap =
                    inputString.chars()
                               .mapToObj(c -> (char) c)
                               .collect(Collectors.groupingBy(word -> word, Collectors.counting()));
        System.out.println(charCountMap); //{ =4, a=3, c=1, C=1, D=1, e=2, f=1, h=1, J=1, n=1, O=1, o=1, p=1, T=1, t=1, v=1, y=1}

    }
}
```

3. **How do you find frequency of each element in an array or a list?**

```
public class SampleTest {

    public static void main(String[] args) {

        List<String> stationeryList = Arrays.asList("Pen", "Eraser", "Note Book", "Pen", "Pencil", "Stapler", "Note Book", "Pencil");

        Map<String, Long> stationeryCountMap =
                stationeryList.stream().collect(Collectors.groupingBy(word -> word, Collectors.counting()));

        System.out.println(stationeryCountMap); //{Pen=2, Stapler=1, Pencil=2, Note Book=2, Eraser=1}
    }

}
```

4. **How do you sort the given list of decimals in reverse order?**

```
public class SampleTest {

    public static void main(String[] args) {
        List<Double> decimalList = Arrays.asList(12.45, 23.58, 17.13, 42.89, 33.78, 71.85, 56.98, 21.12);
        decimalList.stream().sorted(Comparator.reverseOrder()).forEach(System.out::print); //71.85 56.98 42.89 33.78 23.58 21.12 17.13 12.45
    }

}
```

5. **From the given list of integers, print the numbers which are multiples of 5?**

```
public class SampleTest {

    public static void main(String[] args) {
        List<Integer> listOfIntegers = Arrays.asList(45, 12, 56, 15, 24, 75, 31, 89);
        listOfIntegers.stream().filter(i -> i % 5 == 0).forEach(System.out::print); //45 15 75
    }

}
```

6. **Given a list of integers, find maximum and minimum of those numbers?**

```java
public class SampleTest {

    public static void main(String[] args) {
        List<Integer> listOfIntegers = Arrays.asList(45, 12, 56, 15, 24, 75, 31, 89);
        int max = listOfIntegers.stream().max(Comparator.naturalOrder()).get();
        System.out.println("Maximum Element : "+max); //Maximum Element : 89
        int min = listOfIntegers.stream().min(Comparator.naturalOrder()).get();
        System.out.println("Minimum Element : "+min); //Minimum Element : 12
    }

}
```

7. **How do you merge two unsorted arrays into single sorted array using Java 8 streams?**

```java
public class SampleTest {

    public static void main(String[] args) {
        int[] a = new int[] {4, 2, 7, 1};
        int[] b = new int[] {8, 3, 9, 5};
        int[] c = IntStream.concat(Arrays.stream(a), Arrays.stream(b)).sorted().toArray();
        System.out.println(Arrays.toString(c)); //[1, 2, 3, 4, 5, 7, 8, 9]
    }

}
```

8. **How do you merge two unsorted arrays into single sorted array without duplicates?**

```java
public class SampleTest {

    public static void main(String[] args) {
        int[] a = new int[] {4, 2, 5, 1};
        int[] b = new int[] {8, 1, 9, 5};
        int[] c = IntStream.concat(Arrays.stream(a), Arrays.stream(b)).sorted().distinct().toArray(); //[1, 2, 4, 5, 8, 9]
        System.out.println(Arrays.toString(c));
    }

}
```

9. **How do you get three maximum numbers and three minimum numbers from the given list of integers?**

```java
public class SampleTest {

    public static void main(String[] args) {
        List<Integer> listOfIntegers = Arrays.asList(45, 12, 56, 15, 24, 75, 31, 89);

        // 3 minimum Numbers
        listOfIntegers.stream().sorted().limit(3).forEach(System.out::print); //12 15 24

        // 3 Maximum Numbers
        listOfIntegers.stream().sorted(Comparator.reverseOrder()).limit(3).forEach(System.out::print); //89 75 56
    }

}
```

10. **Java 8 program to check if two strings are anagrams or not?**

```java
public class SampleTest {

    public static void main(String[] args) {
        String s1 = "RaceCar";
        String s2 = "CarRace";

        s1 = Stream.of(s1.split("")).map(String::toUpperCase).sorted().collect(Collectors.joining());

        s2 = Stream.of(s2.split("")).map(String::toUpperCase).sorted().collect(Collectors.joining());

        if (s1.equals(s2)) {
            System.out.println("Two strings are anagrams"); //Two strings are anagrams
        } else {
            System.out.println("Two strings are not anagrams");
        }
    }

}
```

11. **Find sum of all digits of a number in Java 8?**

```java
public class SampleTest {

    public static void main(String[] args) {
        int i = 15623;
        Integer sumOfDigits = Stream.of(String.valueOf(i).split("")).collect(Collectors.summingInt(Integer::parseInt));
        System.out.println(sumOfDigits); //17
    }

}
```

12. **Find second largest number in an integer array?**

```java
public class SampleTest {

    public static void main(String[] args) {
        List<Integer> listOfIntegers = Arrays.asList(45, 12, 56, 15, 24, 75, 31, 89);
        Integer secondLargestNumber = listOfIntegers.stream().sorted(Comparator.reverseOrder()).skip(1).findFirst().get();
        System.out.println(secondLargestNumber); //75
    }

}
```

13. **Given a list of strings, sort them according to increasing order of their length?**

```java
public class SampleTest {

    public static void main(String[] args) {
        List<String> listOfStrings = Arrays.asList("Java", "Python", "C#", "HTML", "Kotlin", "C++", "COBOL", "C");
        listOfStrings.stream().sorted(Comparator.comparing(String::length)).forEach(System.out::print);//C C# C++ Java HTML COBOL Python Kotlin
    }

}
```

14. **Given an integer array, find sum and average of all elements?**

```java
public class SampleTest {

    public static void main(String[] args) {
        int[] a = new int[] {45, 12, 56, 15, 24, 75, 31, 89};
        int sum = Arrays.stream(a).sum();
        System.out.println("Sum = "+sum); //Sum = 347
        double average = Arrays.stream(a).average().getAsDouble();
        System.out.println("Average = "+average); //Average = 43.375
    }

}
```

15. **How do you find common elements between two arrays?**

```java
public class SampleTest {

    public static void main(String[] args) {
        List<Integer> list1 = Arrays.asList(71, 21, 34, 89, 56, 28);
        List<Integer> list2 = Arrays.asList(12, 56, 17, 21, 94, 34);
        list1.stream().filter(list2::contains).forEach(System.out::print); //21 34 56
    }

}
```

16. **Reverse each word of a string using Java 8 streams?**

```java
public class SampleTest {

    public static void main(String[] args) {
          String str = "Java Concept Of The Day";
            String reversedStr = Arrays.stream(str.split(" "))
                                       .map(word -> new StringBuffer(word).reverse())
                                       .collect(Collectors.joining(" "));
            System.out.println(reversedStr); //avaJ tpecnoC fO ehT yaD
    }

}
```

17. **Reverse an integer array?**

```java
public class SampleTest {

    public static void main(String[] args) {
        Integer[] array = {1, 2, 3, 4, 5};
        List<Integer> list = Arrays.asList(array);
        Collections.reverse(list);
        Integer[] reversedArray = list.toArray(new Integer[0]);
        System.out.println(Arrays.toString(reversedArray)); //[5, 4, 3, 2, 1]

    }

}
```

18. **How do you find sum of first 10 natural numbers?**

```
public class SampleTest {

    public static void main(String[] args) {
        int sum = IntStream.range(1, 11).sum();
        System.out.println(sum); //55

    }

}
```

19. **Print first 10 even numbers?**

```
public class SampleTest {

    public static void main(String[] args) {
        IntStream.rangeClosed(1, 10).map(i -> i * 2).forEach(System.out::print); //2 4 6 8 10 12 14 16 18 20
    }
}
```

20. **How do you find the most repeated element in an array?**

```
public class SampleTest {

    public static void main(String[] args) {
        List<String> listOfStrings = Arrays.asList("Pen", "Eraser", "Note Book", "Pen", "Pencil", "Pen", "Note Book", "Pencil");
        Map<String, Long> elementCountMap = listOfStrings.stream()
                                                .collect(Collectors.groupingBy(word -> word, Collectors.counting()));
        Entry<String, Long> mostFrequentElement = elementCountMap.entrySet().stream().max(Map.Entry.comparingByValue()).get();
        System.out.println("Most Frequent Element : "+mostFrequentElement.getKey()); //Most Frequent Element : Pen
        System.out.println("Count : "+mostFrequentElement.getValue()); //Count : 3
    }
}
```

21. **Given a list of strings, find out those strings which start with a number?**

```
public class SampleTest {

    public static void main(String[] args) {
        List<String> listOfStrings = Arrays.asList("One", "2wo", "3hree", "Four", "5ive", "Six");
        listOfStrings.stream().filter(str -> Character.isDigit(str.charAt(0))).forEach(System.out::print);//2wo 3hree 5ive
    }
}
```

22. **How do you extract duplicate elements from an array?**

```
public class SampleTest {

    public static void main(String[] args) {
        List<Integer> listOfIntegers = Arrays.asList(111, 222, 333, 111, 555, 333, 777, 222);
        Set<Integer> uniqueElements = new HashSet<>();
        Set<Integer> duplicateElements = listOfIntegers.stream().filter(i -> !uniqueElements.add(i))
                .collect(Collectors.toSet());
        System.out.println(duplicateElements);  //[333, 222, 111]
    }
}
```

23. **Print duplicate characters in a string?**

```java
public class SampleTest {

    public static void main(String[] args) {
        String inputString = "Java Concept Of The Day".replaceAll("\\s+", "").toLowerCase();
        Set<String> uniqueChars = new HashSet<>();
        Set<String> duplicateChars =
                Arrays.stream(inputString.split(""))
                        .filter(ch -> ! uniqueChars.add(ch))
                        .collect(Collectors.toSet());

        System.out.println(duplicateChars); //[a, c, t, e, o]
    }
}
```

24. **Find first repeated character in a string?**

```java
public class SampleTest {

    public static void main(String[] args) {
        String inputString = "Java Concept Of The Day".replaceAll("\\s+", "").toLowerCase();
        Map<String, Long> charCountMap =
                        Arrays.stream(inputString.split(""))
                                .collect(Collectors.groupingBy(word -> word, LinkedHashMap::new, Collectors.counting()));

        String firstRepeatedChar = charCountMap.entrySet()
                                               .stream()
                                               .filter(entry -> entry.getValue() > 1)
                                               .map(entry -> entry.getKey())
                                               .findFirst()
                                               .get();

        System.out.println(firstRepeatedChar); //a
    }
}
```

25. **Find first non-repeated character in a string?**

```java
public class SampleTest {

    public static void main(String[] args) {
        String inputString = "Java Concept Of The Day".replaceAll("\\s+", "").toLowerCase();
        Map<String, Long> charCountMap =
                        Arrays.stream(inputString.split(""))
                                .collect(Collectors.groupingBy(word -> word, LinkedHashMap::new, Collectors.counting()));

        String firstRepeatedChar = charCountMap.entrySet()
                                               .stream()
                                               .filter(entry -> entry.getValue() == 1)
                                               .map(entry -> entry.getKey())
                                               .findFirst()
                                               .get();

        System.out.println(firstRepeatedChar); //j
    }
}
```

26. **How do you get last element of an array?**

```java
public class SampleTest {

    public static void main(String[] args) {
        List<String> listOfStrings = Arrays.asList("One", "Two", "Three", "Four", "Five", "Six");
        String lastElement = listOfStrings.stream().skip(listOfStrings.size() - 1).findFirst().get();
        System.out.println(lastElement); //Six
    }
}
```

27. **Find the age of a person in years if the birthday has given?**

```java
public class SampleTest {

    public static void main(String[] args) {
        LocalDate birthDay = LocalDate.of(1985, 01, 23);
        LocalDate today = LocalDate.now();
        System.out.println(ChronoUnit.YEARS.between(birthDay, today)); //40
    }
}
```

28. **Given a list of integers, find out all the numbers starting with 1 using Stream functions?**

```java
public class SampleTest {

    public static void main(String[] args) {
        List<Integer> myList = Arrays.asList(10, 15, 8, 49, 25, 98, 32);
        myList.stream()
        .map(s -> s + "") // Convert integer to String
        .filter(s -> s.startsWith("1"))
        .forEach(System.out::println); //10 15

    }
}
```

29. **How to find duplicate elements in a given integers list in java using Stream functions?**

```java
public class SampleTest {

    public static void main(String[] args) {
        List<Integer> myList = Arrays.asList(10,15,8,49,25,98,98,32,15);
        myList.stream().distinct().forEach(noDuplicateData -> System.out.print(noDuplicateData)); //10 15 8 49 25 98 32
    }
}
```

30. **Given the list of integers, find the first element of the list using Stream functions?**

```java
public class SampleTest {

    public static void main(String[] args) {
         List<Integer> myList = Arrays.asList(10,15,8,49,25,98,98,32,15);
         myList.stream()
               .findFirst()
               .ifPresent(System.out::println); //10
    }
}
```

31. **Given a list of integers, find the total number of elements present in the list using Stream functions?**

```java
public class SampleTest {

    public static void main(String[] args) {
        List<Integer> myList = Arrays.asList(10,15,8,49,25,98,98,32,15);
        long count =  myList.stream()
                            .count();
        System.out.println(count); //9
    }
}
```

32. **Given a list of integers, find the maximum value element present in it using Stream functions?**

```java
public class SampleTest {

    public static void main(String[] args) {
        int[] arr = {10,15,8,49,25,98,98,32,15};
         int maxdata = Arrays.stream(arr).boxed()
                 .max(Comparator.naturalOrder()).get();
         System.out.println(maxdata);   //98
    }
}
```

33. **Given an integer array nums, return true if any value appears at least twice in the array, and return false if every element is distinct?**

```java
public class SampleTest {

    public static void main(String[] args) {
        int[] nums = {1,2,3,1};
        Set<Integer> setData = new HashSet<>();
        boolean flag =  Arrays.stream(nums)
                       .anyMatch(num -> !setData.add(num));
        System.out.println(flag); //true
    }
}
```

34. **Write a Java 8 program to concatenate two Streams?**

```java
public class SampleTest {

    public static void main(String[] args) {
        List<String> list1 = Arrays.asList("Java", "8");
        List<String> list2 = Arrays.asList("explained", "through", "programs");
        Stream<String> concatStream = Stream.concat(list1.stream(), list2.stream());
        concatStream.forEach(str -> System.out.print(str + " ")); //Java 8 explained through programs
    }
}
```

35. **How to find only duplicate elements with its count from the String ArrayList in Java8?**

```java
public class SampleTest {

    public static void main(String[] args) {
        List<String> names = Arrays.asList("AA", "BB", "AA", "CC");
        Map<String, Long> namesCount = names.stream()
                .collect(Collectors.groupingBy(word -> word, Collectors.counting()))
                .entrySet()
                .stream()
                .filter(entry -> entry.getValue() > 1)
                .collect(Collectors.toMap(Map.Entry::getKey, Map.Entry::getValue));
        System.out.println(namesCount); //{AA=2}
    }
}
```